

The Impact of Autonomous Systems Technology on JPL Mission Software

Dr. Richard J. Doyle

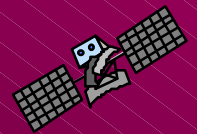
Center for Space Mission Information and Software Systems, and
Information Technologies and Software Systems Division

*Jet Propulsion Laboratory
California Institute of Technology*

24th Annual Software Engineering Workshop
Software Engineering Laboratory
NASA Goddard Space Flight Center



December 1, 1999



RJD.SEL99.1

Autonomy for Future Missions



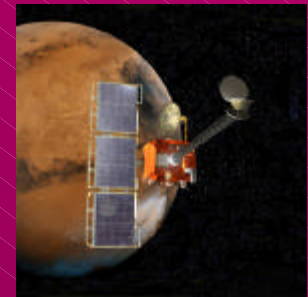
Mars Outposts

- **Remote Science Laboratories**

- Tele-operated or autonomous laboratories in the planetary environment for handling and conducting in situ scientific investigations on collected samples

- **Three scales / resolution**

- remote sensing
- distributed sensing
- point sensing



- **Heterogeneous, cooperating networks**

- distributed networks of sensors, rovers, orbiters, permanent science stations, probes: all of which respond to sensing events, discoveries, changing PI directions, etc., to provide rich presence in Mars environment for science community and public

- **Infrastructure**

- Planetary permanent infrastructure to support series of science and/or commercial missions leading to human presence



JPL

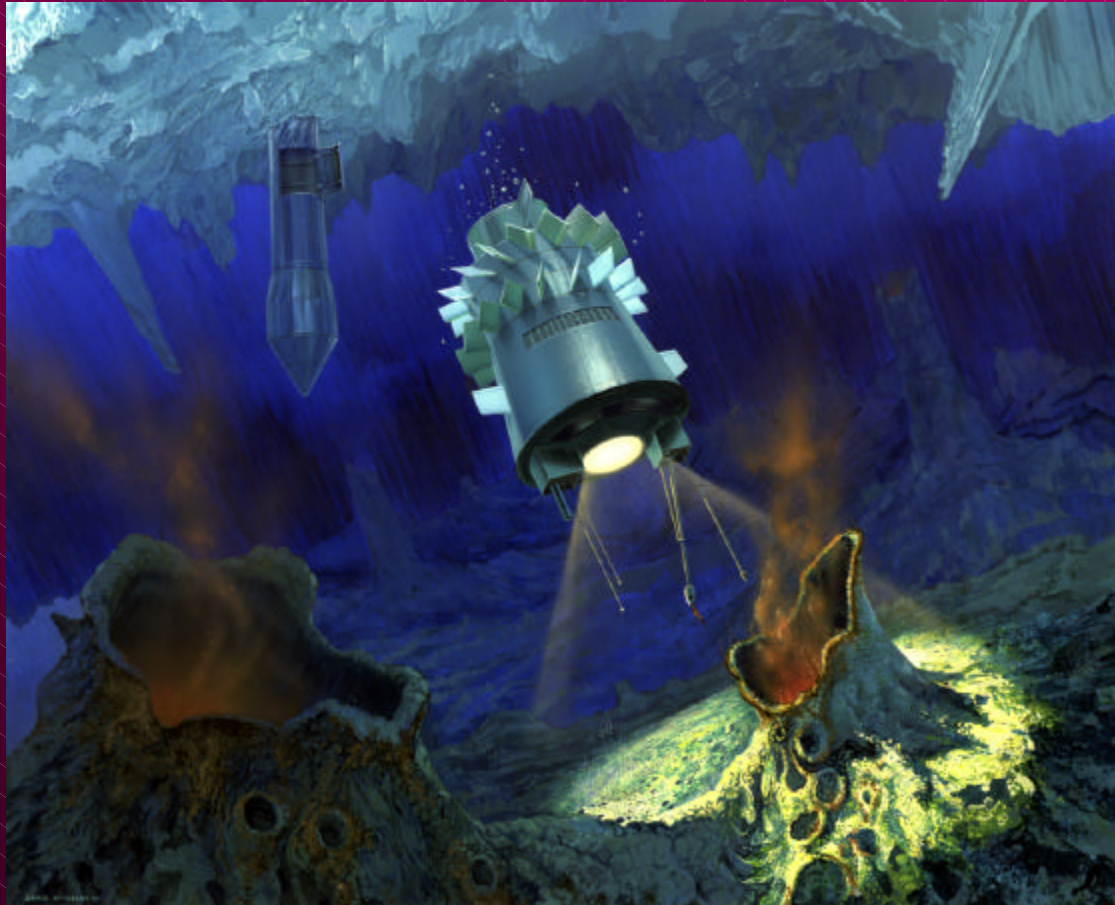
RJD.SEL99.3

Titan Aerobot



- The aerobot conducts in-situ science operations when landed, and wide-area imaging when aloft.
- Archived and learned models of wind patterns assist path planning, enabling near-returns to areas of high scientific interest.

Europa Cryobot / Hydrobot



- Perhaps more than any other, a mission of discovery in a truly alien environment: How to know what to look for? How to recognize it?

The Emergence of Autonomy

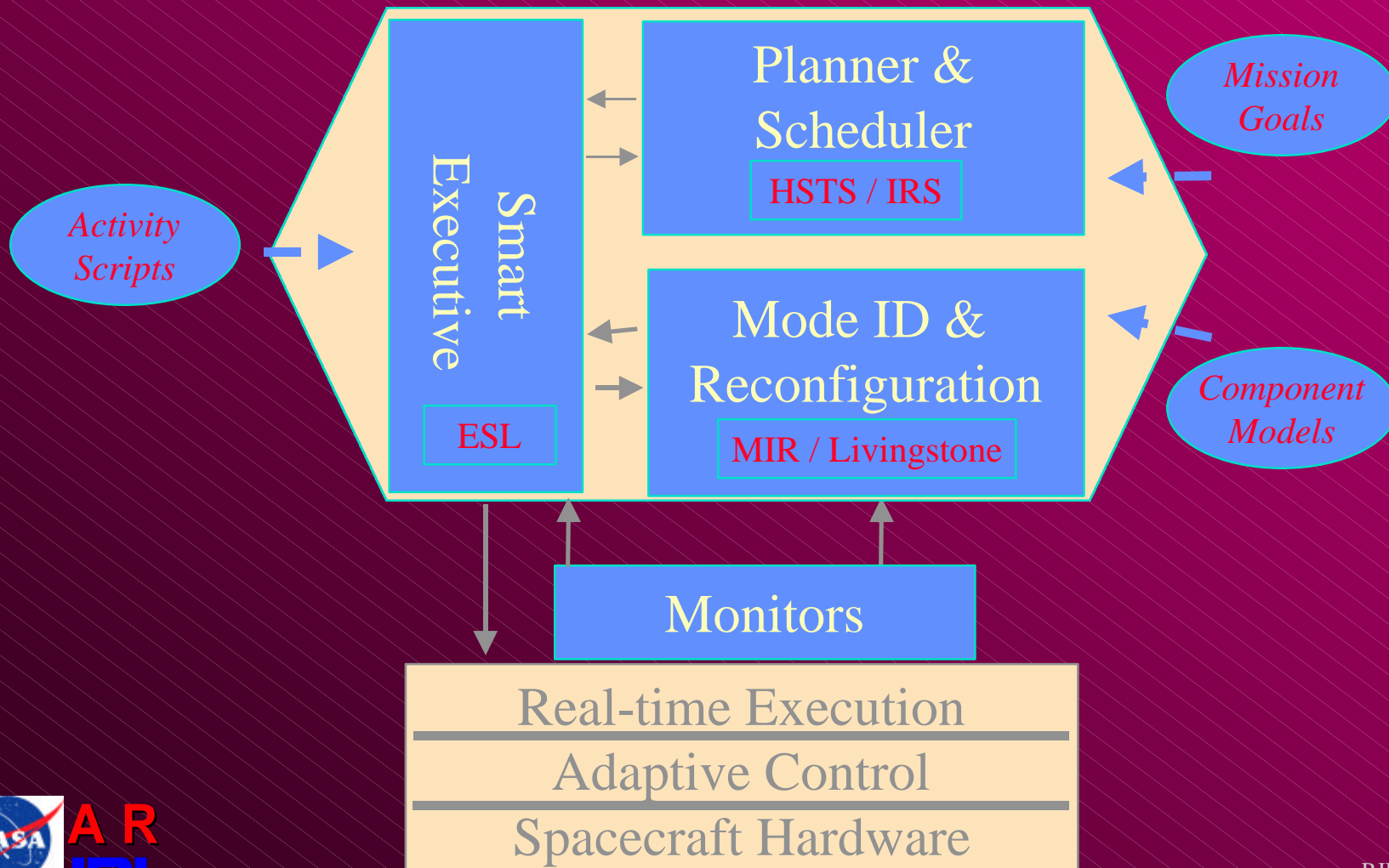


RAX

D. Bernard, P. Nayak et al

Remote Agent eXperiment

Remote Agent Architecture



Closing Loops Onboard

Beacon Operations

Ground assistance invoked with focused report on spacecraft context and history

Planner / Scheduler

Replanning of mission activities around altered resources or functions

Mode Identification & Reconfiguration

Diagnosis of faults and informed selection of recovery actions

Smart Executive

Local retries or alternate, pre-defined activities to achieve same goal

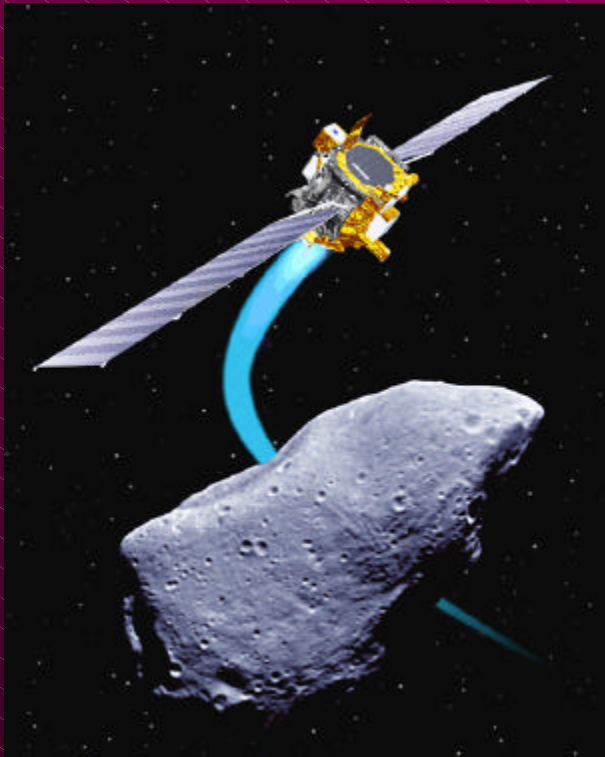
Real-time System

Several layers of onboard recovery provides for unprecedented robustness in achieving mission goals in the face of uncertainty

RAX

Remote Agent eXperiment

New Millennium Flight Experiment



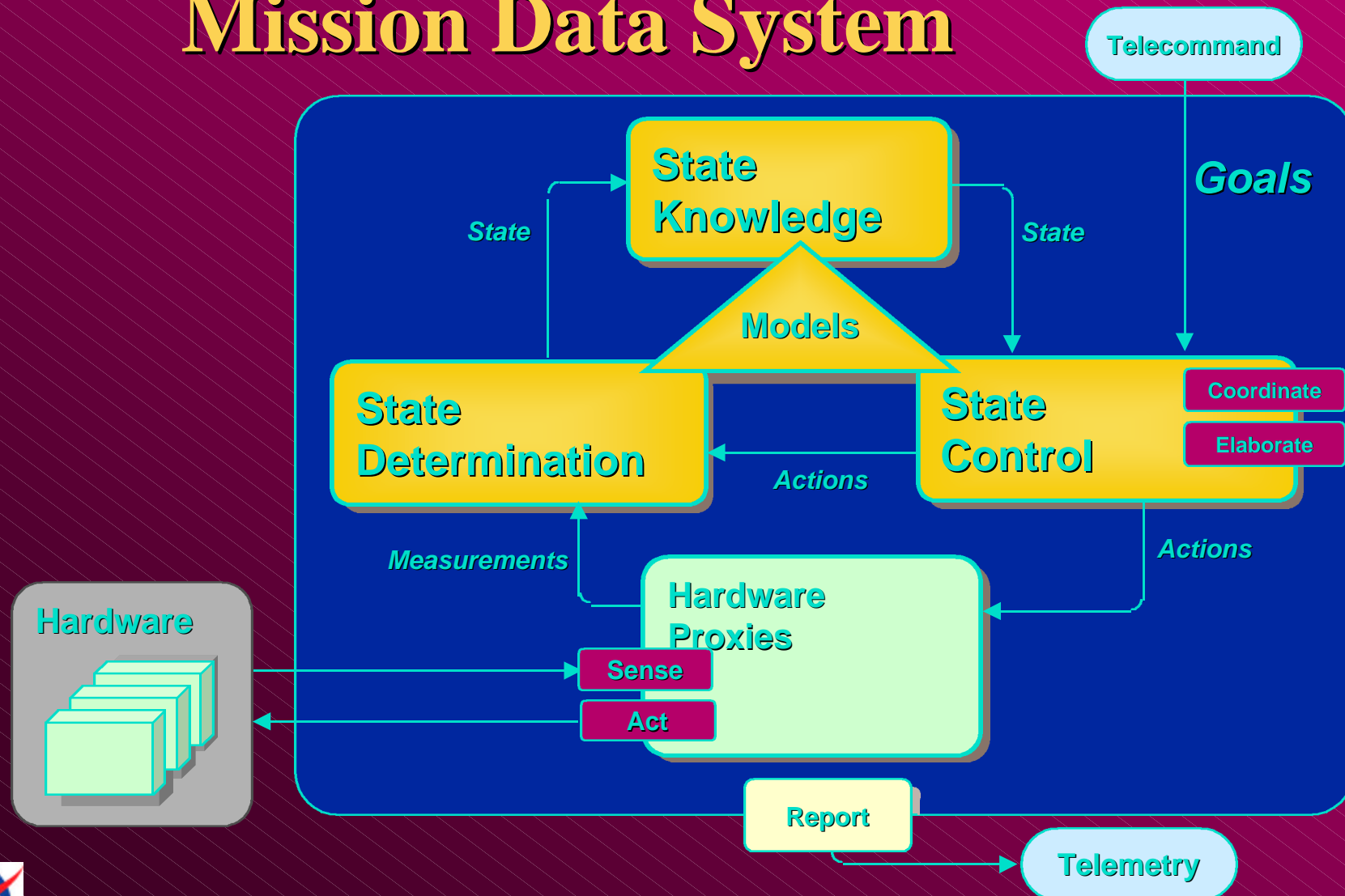
- DS-1 has encountered an asteroid and will encounter a comet.
- Remote Agent Experiment (RAX) achieved 100% of its technology demonstration goals in May '99.
- RAX joined eleven other DS-1 technology experiments such as onboard optical navigation and solar electric propulsion.

Software Engineering Challenges



RJD.SEL99.10

Influence of Remote Agent: Mission Data System




MDS Architectural Themes

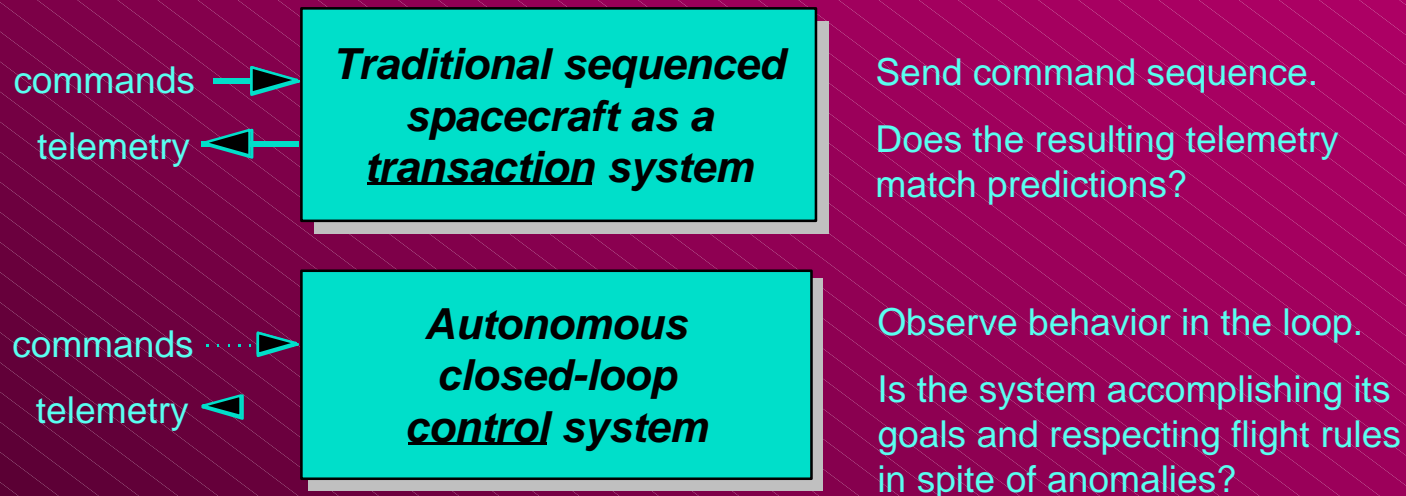
- Unifying state-based paradigm behind all elements
- Extensive and explicit use of models
- Goal-directed operations specifies intent, simplifies workload
- Closed-loop control enables opportunistic science gathering
- Fault protection is natural part of robust control, not an add-on
- Explicit resource management (power, propellant, memory, etc)
- Navigation and attitude control build from common base
- Clean separation of state determination from control
- State uncertainty is acknowledged & used in decision-making
- Clean separation of data management from data transport
- Upward compatibility through careful design of interfaces
- Object-oriented components, frameworks, design patterns



Scalable Autonomy

| <i>Capability</i> | <i>Baseline</i>  | <i>Greater Autonomy</i> |
|-----------------------|--|--|
| Planning & Scheduling | Plans generated and validated on ground; some automation | Plans generated onboard from uplinked goals within s/c and environment context |
| Execution | Highly predictable sequences fully compiled to a timeline | Flexible, deferred commanding; multi-threaded execution; local recovery and cleanup |
| Fault Protection | Fault identification puts spacecraft in safe hold; mission suspended | Model-based diagnosis and recovery for overall s/c state; mission continuation enabled |

Autonomy Software Validation



Key idea: (borrowed from model-based fault diagnosis)

- Do not attempt to enumerate all possible software failures
- Rather, define and identify departures from acceptable bounds on software behavior
- Apply at design, test and run time

Analytic Verification Technology

(Automated Software Engineering Group, NASA Ames)

- Highly autonomous systems typically perform numerous *concurrent* activities, e.g., science observations, instrument calibration, fault monitoring & diagnosis, activity planning, etc.
- Mission systems built upon MDS will employ multi-threaded execution, with an *enormous* space of possible states and paths through those states.
- Concurrent interacting programs are particularly vulnerable to *synchronization bugs* such as race conditions and deadlocks.
- ARC is applying and developing analytic verification technology (a.k.a. “model checking”) to mathematically analyze specifications, code, and models for consistency with requirements and designs:
- At JPL, for Mission Data System (D. Dvorak)
 - At GSFC, for the Advanced Architectures & Agents Group (J. Breed)



Autonomy and Software Engineering

- New critical-path challenges for software engineering are entailed by the autonomy capabilities required for many future NASA missions:
- Verification and validation
 - Reliability
 - Flight / ground architectures
 - Technologies such as auto-code generation
- The specific drivers emerging from the autonomy area are part of a general pattern of increased importance of software engineering to achieve quality mission software

